Maven integrated with Jenkins steps

Creating jar file steps

CI/CD pipeline steps

Types of pipelines

Types of builds

Deploy java-based projectd

* **Creating jar file:**
* Install java
* To check the java installed or not (java -version)
* Install git
* To check the git installed or not (git --version)
* Install maven (sudo wget url -to extract the tar file - tar xvf filename)
* To check the maven installed or not (mvn --version)
* Cd filename
* Ls
* Cd bin/
* Ls
* Pwd (to set the path) copy the path
* Sudo vi /etc/profile (m2-HOME=paste the path)
* Path=$path:paste the path)
* Copy the two paths in the gitbash
* Go to github and paste the url with (git clone repo url)
* Ls (our project source code visible here)
* Cd file name of source code
* Give ls (pom.xml file visible)
* Mvn package
* Give ls (target folder visible here)
* Go to the target directory (cd target/)
* Ls (war file visible here)
* **Maven goals:**

1. Mvn clean
2. Mvn validate
3. Mvn compile
4. Mvn test
5. Mvn package
6. Mvn verify
7. Mvn install
8. Mvn deploy

* **Jenkins:**
* **CI/CD**: is a set of practices that automate the software development and deployment process.
* **CI:** is a software development practice that involves frequently merging code changes into a central repository and then running automated builds and tests.
* Early bug detection
* Improved code quality
* Faster development
* Reduced integration issues
* Quick feedback

Code is automatically tested and ready to deploy to production at any time.

* **Installation Jenkins: there are many ways we are using these two ways**.

1. Deploy Jenkins war file into servlet container. (install tomcat-install java-war file deployed into tomcat server)
2. Installing Jenkins as a service.

**Steps: deploy war file into servlet container (first method)**

Create one ec2 instance with Tomcat

Connecting with gitbash

Install java (command take from official website of amazon Linux)

Install tomcat (go to official website install tar file) (wget paste tar url)

Unzip the tar file (tar xvf url)

Cd filename

Cd bin/

Pwd (copy the path)

Open vi /etc/profile

Startup.sh

Copy the pubilc ip and paste it in google with :8080

Go to google and take the war file of jenkins

Wget paste here link

Copy the public ip and paste it in googole :8080/jenkins

Unlock jenkins(copy root link and paste it in git bash get the administartor password)

Installed suggested plugins

Create username and password

Finish

Start using jenkins

**2nd way using jenkins installation:**

Go to ec2 instance create jenkins server

Launch instance

Coonecting with gitbash

Sudo su

Go to google and installation jenkins

Click on linux

sudo wget -O /etc/yum.repos.d/jenkins.repo \  
 <https://pkg.jenkins.io/redhat-stable/jenkins.repo>

sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

sudo yum upgrade

sudo yum install java-1.8.0-amazon-corretto (install java)  
sudo yum install jenkins  
sudo systemctl daemon-reload

sudo systemctl enable jenkins

sudo systemctl start jenkins

sudo systemctl status jenkins

Copy public ip of jenkins server and paste it in google with :8080

Give administrator password

Installed plugind

Username password

Start using jenkins

How to integrate and configure maven with Jenkins:

Go to manage jenkins

Go to tools

Add maven

Enter maven name

Apply and save

**Java based application deployed:**

Click on new item (enter project name)

Click freestyle

Click on ok

Give description

Go to github and copy url of repo

Click on git and paste repo url

Click on build steps

Click on invoke top level maven targets

Enter name (maven)

Give goal (package)

Click on apply and save

Click on build now

Note : we click on build now but in real time it is automatically will done how

Click on configure

**Build triggers under last 4 we can use in real time**

1. Gitwebhooks trigger:when developer change or modify the code in github it will automatically trigger the job.

Go to github and go to settings go to webhooks left side and enter payload url (it means jenkins url public ip and 8080/github-webhook/) and content type (application/json)

Let me select individual events click on pushes and click on add event. Refresh it updated.

Go to Jenkins job click o configure click on github hook trigger for gitscm polling. click on apply and save.

1. Build after other projects are built build periodically:

It triggered after build it starts test and after it starts a deploy.

1. Poll scm:
2. build periodically:based on time 12 ,1,2.

* **In pipeline we have two types:**

1. Scripted pipeline:

* Scripted pipeline is written in groovy language. Groovy is a scripting language used to define Jenkins pipeline. Allowing for more flexibility and control over the flow of pipeline execution.
* Scripted pipeline starts with node keyword.

1. Declarative pipeline:

* It is more structure and simple syntax for defining Jenkins pipeline.
* Declarative pipeline starts with pipeline keyword.
* Example:
* Pipeline {
* Agent any
* Stages{
* Stage {
* Steps {
* Echo "hello world”
* }
* }
* }
* }
* **Build trigger:**
* Build triggers in Jenkins are mechanisms that automatically start a build based on creation events or conditions.
* **Types of build triggers:**
* **Webhook:**

Whenever developers commit the code, it will automatically trigger the job.

* **Poll SCM:** (Source Control Management)

When we change the code in a particular time, it will automatically trigger the job.

* **Build periodically:**

It will gets trigger the job, even if the changes are made or not.